SCRIBE

MRINAL JHA

24BM6JP35

(04/03/2025)

1. Introduction

This executive summary presents a detailed overview of fundamental concepts of information theory, decision trees, and probability theory. The discussion covers information gain, decision tree learning, Occam's razor, inductive bias, and Bayesian Probability principles.

2. Gini Index

The Gini Index is another measure of impurity, used in CART (Classification and Regression Trees). The internal working of Gini impurity is also somewhat like the working of entropy in the Decision Tree. In the Decision Tree algorithm, both are used for building the tree by splitting as per the appropriate features but there is quite a difference in the computation of both methods. Gini Impurity of features after splitting can be calculated by using this formula. For a classification problem with k classes, if p_i is the probability class i (with $\sum_{i=1}^{k} p_i = 1$), then the Gini Index is defined as:

$$\text{Gini} = 1 - \sum_{i=1}^{k} p_i^2$$

For the special case of binary classification (*k=2*), let p be the probability of the positive class and $1 - p$ the probability of the negative class. The formula simplifies to:

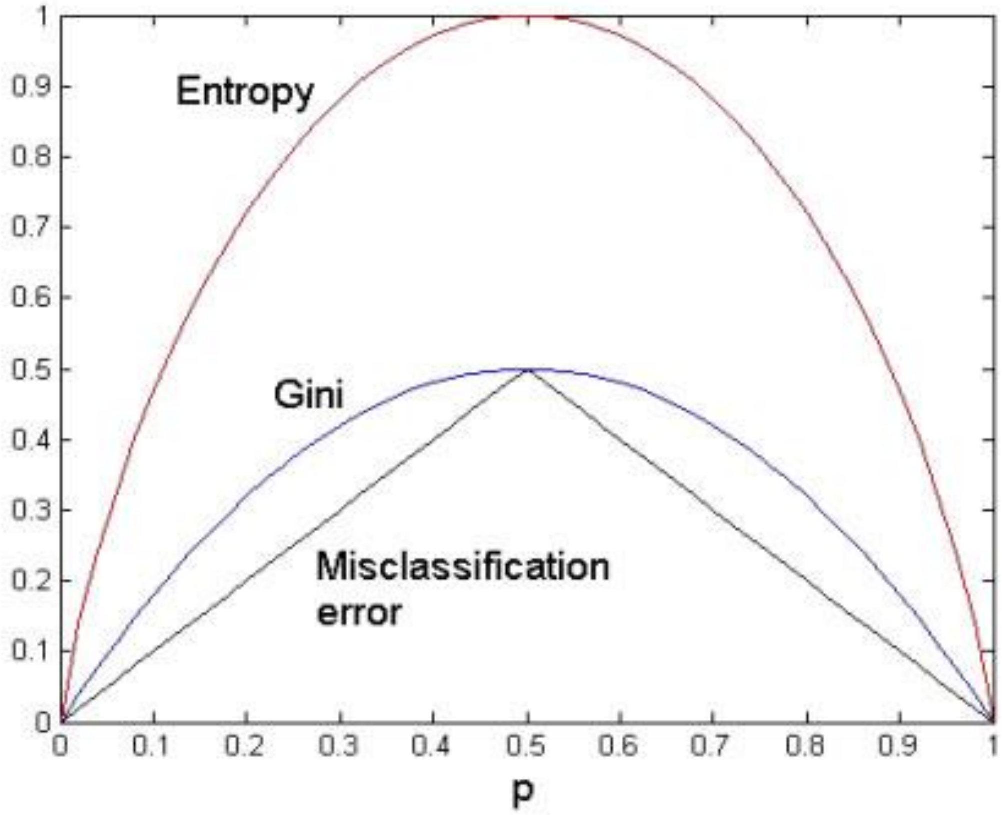
$$\text{Gini} = 1 - (p^2 + (1-p)^2) = 2p(1-p)$$

3. Entropy vs Gini Impurity:

The internal workings of both methods are similar, as they are used for computing the impurity of features after each split**.** However, Gini Impurity is generally more computationally efficient than entropy. The graph of entropy increases up to 1 and then starts decreasing, while Gini Impurity only goes up to 0.5 before decreasing, thus requiring less computational power. The range of entropy is from 0 to 1, whereas the range of Gini Impurity is from 0 to 0.5**.** However, the main reason for Gini Impurity's computational advantage is that it does not involve logarithmic functions, which are more computationally intensive. Therefore, Gini Impurity is often considered more efficient compared to entropy for selecting the best features.

4. Graph of Gini Index for Binary Classification:

Below is the function Gini(p) = 2p (1 − p) for p ∈ [0, 1].



Comparison of Impurity Measures for a binary classification problem

5. Guiding Principle for Model Selection:

A key principle in model selection is Occam's Razor, which suggests that the simplest explanation fitting the data is usually the best. In decision trees, this means favouring smaller, shallower trees that maintain high accuracy, reducing the risk of overfitting and enhancing generalization.

The method used for tree construction—choosing the attribute with the highest information gain or lowest impurity—follows a **greedy** approach, making decisions based on immediate gains without reconsideration. Once a split is made, the algorithm does not backtrack to revise it, leading to an inductive bias toward locally optimal decisions. While this often results in effective trees, it also means that a suboptimal early split cannot be corrected later in the process.

6. Impact of Noise & Missing Data on Decision Trees:

Decision trees are powerful models for classification and regression, but they are sensitive to noise and missing data. If not handled properly, these issues can lead to overly complex trees that overfit training data, resulting in poor generalization. Reduced-error pruning (REP) helps simplify the tree while maintaining accuracy.

6.1 Impact of Noise in Decision Trees:

Noise in data can take many forms, including:

- Incorrect labels: Some data points are misclassified.
- Outliers: Extreme values that do not represent the general trend.
- Random variations: Data inconsistencies that do not follow patterns.

Example: Decision Tree Overfitting Due to Noise:

Let’s say we are classifying students as **"Pass"** or **"Fail"** based on study hours and sleep hours.

With noise, some failing students are mislabelled as passing, and vice versa. A deep tree grows to fit these noisy labels rather than general trends.

Overfitted Tree (With Noise):

```
              Study Hours?
                 /    \
               Yes     No
               /        \
       Sleep?       Extra Activities?
         /  \        /        \
     Fail  Pass   Pass        Fail
```

Problems:

- The tree grows too deep, trying to fit every small fluctuation.
- Some branches exist only because of noise, making the model unreliable.

6.2 Impact of Missing Data in Decision Trees:

If missing data is ignored, decision trees:

- May fail to split correctly, leading to biased trees.
- Can become overly complex, compensating for gaps in data.

6.3 Reduced-Error Pruning (REP):

REP is a technique to simplify a tree by removing unnecessary branches without reducing validation accuracy.

Steps in REP:

- Split Data: Train on one part, validate on another.
- Start at the Leaves: Examine leaf nodes to see if removing them reduces complexity without reducing accuracy.
- Prune If Necessary: Replace branches with majority class if accuracy does not drop.
- Repeat Until No Further Improvement.

7. Conclusion

Decision trees are highly effective for classification and regression tasks, but their performance can be compromised by noise and missing data, leading to overfitting and poor generalization. Noise (such as incorrect labels and outliers) causes unnecessary splits, while missing values can lead to biased decisions or incorrect splits.

To address these challenges, Reduced-Error Pruning (REP) provides a structured approach to simplify decision trees while preserving accuracy. By systematically removing branches that do not improve validation performance, REP ensures that the model remains interpretable and generalizes well to unseen data.

Ultimately, combining pruning techniques with robust handling of missing data leads to more reliable and efficient decision tree models.

8. Probability Theory:

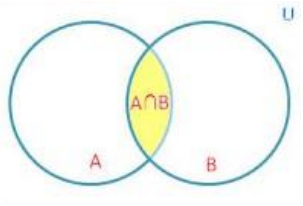
Random Variable: A random variable is a numerical value assigned to the outcomes of a random experiment. It represents uncertainty and is used in probability and statistics to model real-world randomness.

In probability theory, the probability of an event occurring is calculated as:

P(A) = Number of favourable outcomes/Total number of possible outcomes

For the specific case where X represents gender in a sample space, the probability of selecting a female (X=Female) is:

P(X=Female) = Number of females in the sample space/Total number of individuals in sample space.



$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

If events are independent then,

$$P(A \cap B) = P(A) \cdot P(B)$$

8.1 Bayes Rule: Bayes' Rule (or Bayes' Theorem) describes how to update our beliefs about an event based on new evidence.

Formula for Bayes' Rule:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

$$P(A \cap B) = P(A|B) \cdot P(B)$$

$$P(B \cap A) = P(B|A) \cdot P(A)$$

$$\boldsymbol{P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}}$$

8.2 Correlation vs Causality in Bayes' Rule: Bayes' Rule is a **probabilistic framework** that helps update beliefs based on evidence. However, it does not establish **causality**—it only shows **correlation** between events. Understanding the difference is crucial in interpreting results correctly.

8.3 Chain Rule: For any set of events $A_1, A_{2,} \dots, A_n$, the probability of their joint occurrence can be expressed as:

$$\boldsymbol{P(A_1, A_{2,} \dots \dots A_n) = P(A_1) \cdot P(A_2|A_1) \cdot P(A_3|A_2A_1) \dots P(A_n|A_1A_2 \dots A_{n-1})}$$

8.3 Joint Probability Distribution Table: A probability table is used to represent different event combinations, where the probabilities sum to 1.

	A	B	C	Probability
A	1	0	0	0.2
B	0	1	0	0.1
C	1	1	1	0.4

8.4 Challenges with Data Sparsity:

- With **100 data points**, a full probability table would require $2^{\wedge}100 = 10^{\wedge}30$ entries, making it infeasible to compute or store.
- Sparse data makes it difficult to estimate probabilities accurately, as there are too many possible combinations and insufficient observations.

8.4 Approaches to Handle Sparsity

- **Smart Estimation:** Use techniques like smoothing to adjust probability estimates when data is sparse.
- **Smarter Table Representation:** Rather than storing full probability tables, use Bayesian methods and factorized representations to model dependencies efficiently.
- **Bayesian Learning:** Incorporates prior knowledge and observed data to refine probability estimates, reducing overfitting and improving generalization.

8.5 Conclusion: Managing joint probability distributions is essential in probabilistic modelling, but data sparsity presents significant challenges. When datasets are too large to store explicitly or too sparse to estimate probabilities reliably, smart estimation techniques like smoothing and efficient table representations become crucial. Bayesian learning provides a powerful framework for refining probability estimates by incorporating prior knowledge, ensuring more accurate predictions even with limited data.